LA-UR-02-5962

Approved for public release; distribution is unlimited.

Title: SPACE AND TIME CONVERGENCE ANALYSIS OF A CRESTONE PROJECT HYDRODYNAMICS ALGORITHM

Author(s) James R. Kamm, CCS-2
William J. Rider, CCS-2
Jerry S. Brock, X-5

Date: September 2002

Photograph by Chris J. Lindberg

Los Alamos NATIONAL LABORATORY

Space and Time Convergence Analysis of a Crestone Project Hydrodynamics Algorithm

James R. Kamm, William J. Rider, Jerry S. Brock

Los Alamos National Laboratory
Los Alamos, NM 87545 USA

September 25, 2002

LA-UR-02-5962

Contents

Abstract

In this note we describe an analysis of the combined spatial and temporal convergence of a hydrodynamics algorithm in the Crestone Project. Unlike previous analyses, in this study we examine the space-dependent and time-dependent aspects *together*. The analysis of the combined spatial and temporal characteristics of an algorithm leads to a set of nonlinear equations that must be solved numerically. The unknowns in these equations are parameters, including the asymptotic convergence rates, that provide verification metrics. These metrics quantify the performance of the software implementation of the algorithm by gauging the difference between the computed and exact solutions. Restricted to a smooth problem, the design accuracy of the algorithms should be achieved. While we focus on the Euler equations of hydrodynamics in this note, the analysis presented contains the elementary concepts in sufficient detail to apply this technique to a variety of different algorithms or physical circumstances.

1 Introduction

In this study we examine the space-dependent and time-dependent aspects of a hydrodynamics algorithm in the Crestone Project on a smooth[1] problem. Unlike the more common spatial convergence analysis, the combined spatial and temporal analysis leads to a set of nonlinear verification equations that must be solved numerically. The unknowns in this set of equations are various parameters, including the asymptotic convergence rates, that quantify the basic performance of the software implementation of the algorithm.

Whereas in previous reports we examined algorithm behavior on problems with discontinuous solutions (e.g., shocks) [5, 6, 7], in this note we focus on a smooth problem. We do so in order to ascertain the design accuracy of the underlying algorithm, the characteristics of which are known for smooth solutions. Such a demonstration of code convergence is a fundamental aspect of verification analysis, providing evidence with which to establish the correctness of the software implementation of an algorithm.

Essentially all of the descriptions in this note are referenced to the analysis of the Euler equations of compressible hydrodynamics. These equations are a fundamental component of the partial differential equations (PDEs) used in hydrocodes [1]. While we focus on the Euler equations in this note, the analysis we present is applicable to a range of numerical algorithms based on the simultaneous space- and time-discretization of PDEs.

The primary nomenclature and procedures are introduced in §2, which contains detailed descriptions of combined spatial and temporal asymptotic convergence analysis for the Euler equations of hydrodynamics. This discussion

[1] In this note, by a "smooth" problem we mean one that possesses sufficiently many derivatives so that all differentiation operations can be carried out.

provides the framework for the smooth advection problem discussed and analyzed in §3. We summarize the contents of this note in §4.

2 Combined Space and Time Convergence Analysis for the Euler Equations

Whereas many discussions of convergence analysis are couched in general terms applicable to a broad spectrum of PDEs [9, 11], in this note we restrict our attention to the Euler equations for hydrodynamics. We constrain this note to these equations as they form the basis for hydrocodes [1].

2.1 The Euler Equations of Compressible Hydrodynamics

The Euler equations summarize the conservation of mass, momentum, and energy. For a single inviscid, compressible fluid, these equations of two-dimensional motion in Cartesian coordinates are:

$$\begin{array}{lclclc} \frac{\partial\rho}{\partial t} & + & \frac{\partial(\rho u)}{\partial x} & + & \frac{\partial(\rho v)}{\partial y} & = 0\,, \\ \frac{\partial(\rho u)}{\partial t} & + & \frac{\partial\left(\rho u^2+p\right)}{\partial x} & + & \frac{\partial(\rho u v)}{\partial y} & = 0\,, \\ \frac{\partial(\rho v)}{\partial t} & + & \frac{\partial(\rho u v)}{\partial x} & + & \frac{\partial\left(\rho v^2+p\right)}{\partial y} & = 0\,, \\ \frac{\partial(\rho E)}{\partial t} & + & \frac{\partial\left[\rho u\left(E+\frac{p}{\rho}\right)\right]}{\partial x} & + & \frac{\partial\left[\rho v\left(E+\frac{p}{\rho}\right)\right]}{\partial y} & = 0\,, \end{array} \tag{1}$$

where ρ is the mass density, (u,v) are the components of the velocity vector in Cartesian coordinates (x,y), t is the time, $E = e + \frac{1}{2}(u^2+v^2)$ is the specific total energy, e is the specific internal energy, and $p = p(\rho, e)$ is the pressure.

These equations can be written more compactly as

$$\frac{\partial U}{\partial t} + \mathrm{div} F = 0\,, \tag{2}$$

where $U = [\rho, \rho u, \rho v, \rho E]^\top$ is the array of conserved variables, div is the divergence operator, and $F = F(U) = (F_x(U), F_y(U))$ is the flux function.

To obtain numerical solutions, this continuum equation is approximated on a grid that is discrete in both space and time. Furthermore, for the Crestone project code, we consider an Eulerian grid onto which Eq. 2 is discretized. The field of computational fluid dynamic (CFD) is largely devoted to (i) development of appropriate discretizations of this and more general equations, together with (ii) obtaining high fidelity solutions to those discretized equations. Both of these areas are subjects of ongoing and intense research; achievement of either of these missions, in all but the simplest of cases, can by no means be considered a *fait accompli*.

The corresponding solution of the discretized form of Eq. 2 is indicated as $U^l_{i,j}$. This value corresponds to $U(x_i, y_j; t_l)$, the solution at position (x_i, y_j) and time t_l. We say "corresponds to" because the nature of the precise relationship between the discrete solution and the continuous solution depends on the solution algorithm. In the subsequent discussion we assume a *uniform* and *equal* spatial grid with $\Delta x = \Delta y$ and *uniform* and *equal* timesteps Δt.

2.2 Interacting Space and Time Discretizations

The nature of the numerical scheme used to discretize continuous PDEs into a set of discrete equations can affect the nature of the computational error associated with the numerical solution. For example, one can devise numerical schemes for PDEs that produce separate, non-coupled spatial and temporal discretization errors. Modern high-resolution numerical schemes for conservation laws (such as described in [3, 8]), however, may not retain strict separation of spatial and temporal discretizations. Consequently, an interaction of the spatial and temporal discretizations is at play in such methods. Such is the case for the hydrodynamics algorithm of the Crestone project code, which uses a Godunov-type method with Lax-Wendroff time differencing. For such an algorithmn, the temporal dependence is interwoven with the spatial dependence through the self-similar solutions to local Riemann problems.

To characterize the combined spatio-temporal dependence of the error in the solution, we analyze the *average* per-cycle convergence properties by postulating the following error Ansatz:

$$\begin{aligned} ||\xi^* - \xi_i^l||/N_{\Delta t_l} = \mathcal{E}_0 &+ \mathcal{A}\,(\Delta x_i)^p + \mathcal{B}\,(\Delta t_l)^q + \mathcal{C}\,(\Delta x_i)^r\,(\Delta t_l)^s \\ &+ o\Big((\Delta x_i)^p\,, (\Delta t_l)^q\,, (\Delta x_i)^r\,(\Delta t_l)^s\Big)\,, \end{aligned} \tag{3}$$

where ξ is some functional of the solution (e.g, one component of U, say, ρ), ξ^* is the *exact* value, ξ_i^l is the value computed on the grid of spatial zone size Δx_i with timestep Δt_l, $||\cdot||$ is a norm[2], $N_{\Delta t_l}$ is the number of time cycles taken to obtain the solution at the final time, $\mathcal{E}_0$ is the *zeroth-order error*, $\mathcal{A}$ is the *spatial convergence coefficient*, p is the *spatial convergence rate*, $\mathcal{B}$ is the *temporal convergence coefficient*, q is the *temporal convergence rate*, $\mathcal{C}$ is the *spatio-temporal convergence coefficient*, and $r+s$ is the *spatio-temporal convergence rate*.

The error Ansatz in Eq. 3 averages out the location-to-location, cycle-to-cycle dependence of the computed results on Δx and Δt, yielding a measure of typical per-cycle solution algorithm performance. To put this Ansatz into perspective, consider the key aspects of the quantity under consideration, i.e., of the norm of the difference between the exact and computed solution, divided by the number of computational cycles. The solution norm, which is typically a discrete approximation to some integral of the norm's argument, can be interpreted as a spatial averaging operator; that is, the norm quantifies some

[2] In this case, a norm is a mapping of the solution at a specified time on the discrete grid to the non-negative real numbers; see [4] for further details.

mean measure of the spatial behavior of the solution difference. The quotient of this quantity divided by the number of computational cycles likewise provides a temporal averaging operator; unlike the spatial norm, however, this operation produces a mean per-cycle measure.

2.3 Equations for Global Space and Time Convergence with a Known Exact Solution

As discussed in [7], there are several canonical cases, depending on the relative magnitudes of various parameters in Eq. 3. We retain the various parameters characterizing the spatial and temporal convergence in Eq. 3.

We make two additional assumptions prior to describing the technique by which we determine these parameters: (i) we assume that the zeroth-order error is negligible, i.e.,

$$|\,\mathcal{E}_0\,| \ll |\,\mathcal{A}\ (\Delta x_i)^p\ |, |\,\mathcal{B}\ (\Delta t_l)^q\ |, |\,\mathcal{C}\ (\Delta x_i)^r\ (\Delta t_l)^s\ |; \tag{4}$$

and (ii) we presume that we know *a priori* the exact solution ξ^* at any grid location at the desired time. These assumptions imply that there are a total of seven unknowns in Eq. 3: A, p, B, q, C, r, and s. To solve for these quantities, we require seven independent equations. To do so, we obtain computed solutions at the same final time with the following seven combinations of spatial and temporal zoning:

$$\begin{array}{lll} (1) & : & \{\Delta x \quad ,\Delta t\}\,, \\ (2) & : & \{\Delta x/\sigma\ ,\Delta t\}\,, \\ (3) & : & \{\Delta x/\sigma^2,\Delta t\}\,, \\ (4) & : & \{\Delta x \quad ,\Delta t/\tau\}\,, \\ (5) & : & \{\Delta x \quad ,\Delta t/\tau^2\}\,, \\ (6) & : & \{\Delta x/\sigma\ ,\Delta t/\tau\}\,, \\ (7) & : & \{\Delta x/\sigma\ ,\Delta t/\tau^2\}\,, \end{array} \tag{5}$$

where $\sigma > 1$ is the ratio of the spatial grid sizes, and $\tau > 1$ is the ratio of the temporal grid sizes. This set of zonings is neither unique nor demonstrably optimal for obtaining solutions of Eq. 3; however, it does provide a sufficient set of independent information with which to obtain solutions for the unknowns in Eq. 3. Together with the no-zeroth-order-error assumption, the computed solutions on these space-time grids satisfy the following equalities at the (identical) final time:

$$\begin{array}{l} 0 = f_1 = -||\xi^* - \xi_1||/N_c \ + \mathcal{A}\ (\Delta x_c)^p + \mathcal{B}\ (\Delta t_c)^q + \mathcal{C}\ (\Delta x_c)^r\ (\Delta t_c)^s\,, \\ 0 = f_2 = -||\xi^* - \xi_2||/N_c \ + \mathcal{A}\ (\Delta x_m)^p + \mathcal{B}\ (\Delta t_c)^q + \mathcal{C}\ (\Delta x_m)^r\ (\Delta t_c)^s\,, \\ 0 = f_3 = -||\xi^* - \xi_3||/N_c \ + \mathcal{A}\ (\Delta x_f)^p + \mathcal{B}\ (\Delta t_c)^q + \mathcal{C}\ (\Delta x_f)^r\ (\Delta t_c)^s\,, \\ 0 = f_4 = -||\xi^* - \xi_4||/N_m + \mathcal{A}\ (\Delta x_c)^p + \mathcal{B}\ (\Delta t_m)^q + \mathcal{C}\ (\Delta x_c)^r\ (\Delta t_m)^s\,, \\ 0 = f_5 = -||\xi^* - \xi_5||/N_f \ + \mathcal{A}\ (\Delta x_c)^p + \mathcal{B}\ (\Delta t_f)^q + \mathcal{C}\ (\Delta x_c)^r\ (\Delta t_f)^s\,, \\ 0 = f_6 = -||\xi^* - \xi_6||/N_m + \mathcal{A}\ (\Delta x_m)^p + \mathcal{B}\ (\Delta t_m)^q + \mathcal{C}\ (\Delta x_m)^r\ (\Delta t_m)^s\,, \\ 0 = f_7 = -||\xi^* - \xi_7||/N_f \ + \mathcal{A}\ (\Delta x_m)^p + \mathcal{B}\ (\Delta t_f)^q + \mathcal{C}\ (\Delta x_m)^r\ (\Delta t_f)^s\,. \end{array} \tag{6}$$

In these expressions, $\Delta x_c \equiv \Delta x$ is the coarse spatial grid size, $\Delta x_m \equiv \Delta x/\sigma$ is the medium spatial grid size, and $\Delta x_f \equiv \Delta x/\sigma^2$ is the fine spatial grid size; similarly, $\Delta t_c \equiv \Delta t$ is the coarse timestep, $\Delta t_m \equiv \Delta t/\tau$ is the medium timestep, and $\Delta t_f \equiv \Delta t/\tau^2$ is the fine timestep. Also, N_c, N_m, and N_f represent the number of time cycles involved in computing the solutions with the coarse, medium, and fine timesteps, respectively.

This set of seven equations depends on the variable $\mathbf{a} \equiv [a_1, \ldots, a_7]^\top \equiv [\mathcal{A}, p, \mathcal{B}, q, \mathcal{C}, r, s]^\top$:

$$\begin{aligned}
0 = f_1 &= -||\xi^* - \xi_1||/N_c + a_1\, (\Delta x_c)^{a_2} + a_3\, (\Delta t_c)^{a_4} + a_5\, (\Delta x_c)^{a_6}\, (\Delta t_c)^{a_7}\,, \\
0 = f_2 &= -||\xi^* - \xi_2||/N_c + a_1\, (\Delta x_m)^{a_2} + a_3\, (\Delta t_c)^{a_4} + a_5\, (\Delta x_m)^{a_6}\, (\Delta t_c)^{a_7}\,, \\
0 = f_3 &= -||\xi^* - \xi_3||/N_c + a_1\, (\Delta x_f)^{a_2} + a_3\, (\Delta t_c)^{a_4} + a_5\, (\Delta x_f)^{a_6}\, (\Delta t_c)^{a_7}\,, \\
0 = f_4 &= -||\xi^* - \xi_4||/N_m + a_1\, (\Delta x_c)^{a_2} + a_3\, (\Delta t_m)^{a_4} + a_5\, (\Delta x_c)^{a_6}\, (\Delta t_m)^{a_7}\,, \\
0 = f_5 &= -||\xi^* - \xi_5||/N_f + a_1\, (\Delta x_c)^{a_2} + a_3\, (\Delta t_f)^{a_4} + a_5\, (\Delta x_c)^{a_6}\, (\Delta t_f)^{a_7}\,, \\
0 = f_6 &= -||\xi^* - \xi_6||/N_m + a_1\, (\Delta x_m)^{a_2} + a_3\, (\Delta t_m)^{a_4} + a_5\, (\Delta x_m)^{a_6}\, (\Delta t_m)^{a_7}\,, \\
0 = f_7 &= -||\xi^* - \xi_7||/N_f + a_1\, (\Delta x_m)^{a_2} + a_3\, (\Delta t_f)^{a_4} + a_5\, (\Delta x_m)^{a_6}\, (\Delta t_f)^{a_7}\,,
\end{aligned} \tag{7}$$

where the indices correspond to the numbering in Eq. 5. To obtain solutions to this set of nonlinear equations $\mathbf{f}(\mathbf{a}) = \mathbf{0}$, we use a modified line-search based Newton's method [10]. It is straightforward to obtain closed-form expressions for the elements of the corresponding Jacobian $\mathcal{J}$, with elements $\mathcal{J}_{i,j} \equiv \partial f_i/\partial a_j$, the inverse of which is typically evaluated numerically in Newton's method-based routines.

To obtain solutions to Eq. 7, one must: (i) obtain numerical solutions from the hydrocode to the fixed final time using the spatial and temporal grids specified, and (ii) assign an initial guess for the array $\mathbf{a}$ that is within the domain of convergence of the iteration. The former is merely a matter of relatively meager computer resources, whereas the latter requires some *a priori* knowledge of the algorithm of interest; the obvious choice for initial guess consists of the algorithm's theoretical convergence rate together with, say, estimates of the convergence parameter from a purely spatial convergence analysis. There are two additional practical concerns in this procedure. The first is that of properly assigning the parameters associated with convergence in the Newton's method implementation; this is a matter requiring some user interaction, using guidance from the Newton's method routine. The second is that of resolving any multiple solutions obtained to this set of nonlinear equations; response to this issue will vary on a case-by-case basis.

3 Convergence Results for a Smooth Problem

To test the underlying hydrodynamics algorithms on smooth problems we consider a problem that possesses smooth initial conditions and that is evolved to a point prior to the development of any discontinuities. Whereas one approach

2-D Sinusoidal Density Advection Problem Initial Conditions

γ	ρ	p	e	u	v
1.4	$2+\sin 2\pi x\ \sin 2\pi y$	1.0	$2.5/\left(2+\sin 2\pi x\ \sin 2\pi y\right)$	1.0	1.0

Table 1: Initial values of the adiabatic exponent γ, nondimensional density ρ, pressure p, SIE e, x-velocity u, and y-velocity v for the 2-D sinusoidal density advection problem.

would be to use the Method of Manufactured Solutions [12], we instead use known solutions of the Euler equations; our approach obviates the need to deal with extraneous source terms in the equations and possible modifications of the solution algorithm. The numerical solutions that we obtain with different spatial and temporal meshes are compared with the exact solution at identical final times. The convergence properties of the coded algorithm are then inferred following the procedure outlined above.

3.1 2-D Sinusoidal Advection Problem Initial Conditions

The two-dimensional, planar geometry initial conditions for this problem consist of a sinusoidal distribution of density with initially constant and uniform pressure, thermodynamically consistent specific internal energy (SIE), and uniform non-zero velocity (u_0, v_0). The equation of state (EOS) is chosen to be a polytropic gas with adiabatic exponent $\gamma = 1.4$. With periodic boundary conditions, this configuration advects the sinusoidal density and SIE distributions, which remain unperturbed, through the computational mesh. If we write the initial conditions as $f(x,y)$, then the solution at any time t is given by $f(x - u_0\,t, y - v_0\,t)$. More precisely, the domain of interest is assigned to be the square of unit dimension centered at the origin in Cartesian geometry, i.e., $\{(x,y) : -1/2 \leq x \leq 1/2 \text{ and } -1/2 \leq y \leq 1/2\}$. The initial conditions for this problem are given in Table 1.

One caveat to this problem is that it tests only the linear fields in the governing equations. We are investigating the smooth simple wave problem proposed by Cabot [2] as a complementary problem that exercises the nonlinear fields of Eq. 2.

3.2 Results for a Crestone Project Hydrodynamics Algorithm

Calculations of all problems were carried out with a Crestone Project code on uniform grids consisting of 32×32, 64×64, 128×128, and 256×256 zones. Timesteps of 1/1600, 1/3200, 1/6400, 1/12800 were used. Thus, both the subsequent spatial and temporal zone sizes used in computing the convergence properties were a factor of two smaller, i.e., $\sigma = \tau = 2$ in the nomenclature of the previous section. These timesteps are well below the CFL limit for this set of calculations: for the parameters chosen, the maximum sound speed is 1.183,[1] so, according to Eq. 8, the maximum CFL value attained is

$$C_{max} = (\max\{|u|, |v|\} + c_{max})\ \Delta t_{max}/\Delta x_{min} \doteq 0.044 < 1\,. \tag{8}$$

To ensure that the uniform spatial grid was retained, no adaptive mesh refinement (AMR) capabilities were utilized in these simulations. These sets of calculations provide data with which verification metrics, i.e., the quantities characterizing the spatial and temporal convergence of the fundamental hydrodynamics algorithm, were obtained.

As in all such verification analyses, it must be stressed that the solution values must be compared at *identical locations* in space at *exactly the same time*. Simple interpolation of solutions provides values at identical spatial locations [5, 6], and the choice of fixed timesteps allows solutions to be obtained at the identical final time, $t = 0.1$.

The results of the suite of calculations conducted on 32×32, 64×64, and 128×128 grids are presented in Table 2, and results based on 64×64, 128×128, and 256×256 grids are given in Table 3. While the spatial and combined spatio-temporal convergence rates (i.e., p and $r + s$) are approximately two in all cases, at the finer temporal resolutions (e.g., for $\Delta t = 1/6400$ and $1/12800$) there appears to be a marked decrease in the temporal order of the algorithm (i.e., q), to approximately first order. There are two obvious explanations for this behavior: (1) the nonlinear solver used in the convergence analysis, described in §2.3, has converged onto this solution, while a solution with $q \approx 2$ may still exist but was not found, and (2) the behavior of the temporal integrator degrades at smaller timesteps, which correspond to smaller CFL numbers (see Eq. 8). The former issue could be examined with numerical methods that identify *all* roots of nonlinear systems of equations; the results of such an investigation may shed light on the likelihood of the latter possibility. At this point, however, it is unclear which of these explanations is most plausible.

There are possible implications of these preliminary findings for the interaction of the hydrodynamics algorithm solution with the Crestone Project adaptive mesh refinement (AMR) techniques. This study did not exercise any AMR capabilities, which can be used to locally refine the spatial discretization of the solution. If the *global* timestep for the entire computational mesh is assigned so that it does not violate the CFL condition in the smallest zone, then the corresponding CFL number in the largest zone may well be significantly smaller. Our limited findings suggest the possibility of degradation of the temporal order of accuracy to approximately one at very small CFL numbers for the advection

[1] The maximum soundspeed is given by $c_{max} = \sqrt{\gamma\, p_{max}/\rho_{min}} \doteq 1.183$.

$t = 0.1$ Convergence Results for 32×32, 64×64, and 128×128 Grids

Δt	N_c	$\mathcal{A} \times 10^2$	p	$\mathcal{B} \times 10^2$	q	$\mathcal{C} \times 10^2$	r	s
1/1600 1/3200 1/6400	160 320 640	1.00	1.90	0.67	1.95	1.00	0.90	0.90
1/3200 1/6400 1/12800	320 640 1280	1.00	2.00	0.24	1.89	1.02	1.00	1.00
1/6400 1/12800 1/25600	640 1280 2560	1.00	2.00	-5.82	1.13	1.08	1.08	1.00
1/12800 1/25600 1/51200	1280 2560 5120	1.00	2.00	-6.43	1.13	1.04	1.01	1.00

Table 2: Convergence quantities for the smooth advection problem calculated with 32^2, 64^2, and 128^2 zones on the unit square with the indicated timesteps Δt and number of computational cycles N_c. The other parameters are defined in the text.

problem considered. If this behavior were to be true in general, then the possibility exists that the temporal integration errors in large zones could be at (approximately) first order while the corresponding temporal integration errors in small zones may be (approximately) second order. Of course, the cumulative effect of this phenomenon on overall solution accuracy would be problem dependent. Given the speculative nature of this observation and the possibilities raised in the preceding paragraph, further investigation and elucidation of the details of space and time convergence analysis seems warranted.

$t = 0.1$ Convergence Results for 64×64, 128×128, and 256×256 Grids

Δt	N_c	$\mathcal{A}\times 10^2$	p	$\mathcal{B}\times 10^2$	q	$\mathcal{C}\times 10^2$	r	s
1/1600 1/3200 1/6400	160 320 640	1.00	2.00	0.78	1.97	1.02	1.01	1.00
1/3200 1/6400 1/12800	320 640 1280	1.00	2.00	0.28	1.89	1.05	1.01	1.00
1/6400 1/12800 1/25600	640 1280 2560	1.00	2.00	-2.48	1.56	1.07	1.01	1.00
1/12800 1/25600 1/51200	1280 2560 5120	1.00	2.00	-6.43	1.13	1.08	1.02	1.00

Table 3: Convergence quantities for the smooth advection problem calculated with 64^2, 128^2, and 256^2 zones on the unit square with the indicated timesteps Δt and number of computational cycles N_c. The other parameters are defined in the text.

4 Summary

In this note we have performed convergence analysis in both space and time on a smooth problem for a hydrodynamics algorithm in the Crestone Project. The fundamental assumption of this analysis (Eq. 3) is that the mean per-cycle error in the computed solution varies as a polynomial in the computational cell size and computational timestep, with the exponents in this expression being the convergence rates. Unlike the direct evaluation of convergence properties for the standard spatial convergence analysis, the combined space-time analysis requires the numerical solution of a set of nonlinear verification equations. Obtaining solutions to this set of equations, while straightforward, is more involved than directly obtaining the results for the space-only or time-only convergence behavior.

The descriptions of the assumptions underlying this analysis are given in §2, where the equations governing this approach are derived. An application of this analysis is provided using the smooth advection problem specified in §3.1. The results of our study, described in §3.2, demonstrate that the underlying advection algorithm is indeed second order in space. In contrast, the temporal integrator exhibits second order behavior for coarser temporal grids but transitions to nearly first order for finer timesteps. The combined spatio-temporal rate of convergence is nominally two at all resolutions considered.

Using the methodology presented herein, one can implement a combined space-and-time convergence analysis procedure by which to verify software implementations of discrete numerical algortihms used in obtaining space- and time-dependent solutions of PDEs.

Acknowledgements

The authors wish to acknowledge ongoing and enlightening discussions with Michael Clover and Rob Lowrie on code verification issues. This work was performed at Los Alamos National Laboratory, which is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36.

References

[1] D. J. Benson, Computational Methods in Lagrangian and Eulerian Hydrocodes, *Computer Methods Applied Mechanics Eng.* **99**, pp. 235–394 (1992).

[2] W. Cabot, *Simulations of a one-dimensional simple wave*, unpublished manuscript (2002).

[3] P. Colella, E. G. Puckett, *Modern Numerical Methods for Fluid Flow*, unpublished manuscript, available at `http://www.amath.unc.edu/Faculty/minion/class/C_P_Notes.pdf` (1998).

[4] G. Dahlquist, Å. Björk, *Numerical Methods*, Prentice-Hall, Englewood Cliffs, NJ (1974).

[5] J. R. Kamm, W. J. Rider, *2-D Convergence Analysis of the RAGE Hydrocode*, Los Alamos National Laboratory Report LA-UR-98-3872 (1998).

[6] J. R. Kamm, W. J. Rider, *Verification Analysis of the Direct Eulerian 2-D RAGE Hydrodynamics Algorithm*, Los Alamos National Laboratory Report LA-UR-99-5234 (1999).

[7] J. R. Kamm, W. J. Rider, J. S. Brock, *Consistent Metrics for Code Verification*, Los Alamos National Laboratory Report LA-UR-02-3794 (2002).

[8] R. J. Leveque, *Finite Volume Methods for Hyperbolic Problems*, Cambridge University Press, Cambridge, UK (2002).

[9] W. L. Oberkampf, T. G. Trucano, *Verification and Validation in Computational Fluid Dynamics*, Sandia National Laboratories Report SAND2002-0529 (2002).

[10] W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling, *Numerical Recipes, The Art of Scientific Computing*, Cambridge University Press, Cambridge, UK (1992).

[11] P. J. Roache, *Verification and Validation in Computational Science and Engineering*, Hermosa Publishers, Albuquerque, NM (1998).

[12] K. Salari, P. Knupp, *Code Verification by the Method of Manufactured Solutions*, Sandia National Laboratories Report SAND2000-1444 (2000).